

*Una buena ingeniería de software
requiere la diferenciación entre la
especificación y la implementación.*

INGENIERIA DE SOFTWARE

Primer Avance

García Juárez Leonel Alexis

Medina Avendaño Griselda

Ramírez Ramírez Ramiro

Índice general

0.1. Introducción	5
1. Propósito	6
2. Ámbito del Proyecto	7
3. Personal Involucrado	8
4. Factibilidad	9
5. Viabilidad	10
6. Mapa de Navegación	11
7. Requisitos Funcionales	12
8. Interfaces	22
9. Estándar de Codificación	33
9.1. Inicialización	33
9.2. Variables locales	33
9.3. Variables globales	33
9.4. Constantes	33
9.5. Comentarios de inicio o encabezado	34
9.6. Comentarios en línea	34
9.7. Sangrías y tabulaciones	34
9.8. Interfaces	34
10. Diagramas de casos de uso	36
11. Diccionario de datos	54
12. Diseño Lógico	57
13. Diseño Físico	60

Índice de figuras

3.1. Roles de usuario	8
6.1. Mapa de navegación	11
7.1. Requisito Funcional 1	12
7.2. Requisito Funcional 2	13
7.3. Requisito Funcional 3	14
7.4. Requisito Funcional 4	15
7.5. Requisito Funcional 5	16
7.6. Requisito Funcional 6	17
7.7. Requisito Funcional 7	18
7.8. Requisito Funcional 8	19
7.9. Requisito Funcional 9	20
7.10. Requisito Funcional 10	21
8.1. Login	22
8.2. Pantalla Principal	23
8.3. Productos	24
8.4. Nuevo Producto	25
8.5. Modificar Producto	26
8.6. Eliminar Producto	27
8.7. Inventario	28
8.8. Ventas	29
8.9. Reporte	30
8.10. Corte	31
8.11. Usuarios	32
10.1. CU Login	37
10.2. Diagrama de CU Ventas	38
10.3. CU Ventas	39
10.4. Diagrama de CU Nuevo Producto	40
10.5. CU Nuevo Producto	41
10.6. Diagrama de CU Modificar Producto	42
10.7. CU Modificar Producto	43
10.8. Diagrama de CU Eliminar Producto	44
10.9. CU Eliminar Producto	45
10.10. Diagrama de CU Inventario	46
10.11. CU Inventario	47

10.12	Diagrama de CU Corte	48
10.13	CU Corte	49
10.14	Diagrama de CU Reporte	50
10.15	CU Reporte	51
10.16	Diagrama de CU Usuarios	52
10.17	CU Uusario	53
11.1.	Productos	54
11.2.	Ventas	55
11.3.	Categorías	55
11.4.	Empleado	55
11.5.	Pertenece	55
11.6.	Registro	56
12.1.	Diagrama ERD	58
12.2.	Esquema Relacional	59

.

0.1. Introducción

Este proyecto ha sido elaborado dada la necesidad de automatizar un proceso de ventas de una ferretería, la cual requiere generar informes exactos sobre las ventas diarias, además de que requiere de llevar un control de inventario de productos para poder supervisar el stock disponible de los mismos, además se pretende que a través de este sistema los empleados mejoren sus tiempos de atención al cliente en ventas ya que evitarían hacer cuentas de forma manual.

El sistema de software que implementaremos facilitará el manejo de ventas, inventario, ingreso de productos, etc ya que esto se realizará de manera sistematizada ya que actualmente se realiza de manera manual, por lo anterior de este sistema ayudaría a visualizar el rendimiento de la empresa tomando en cuenta las ventas generadas por el sistema.

Este proyecto será realizado con el lenguaje de programación Java, además de que teniendo en cuenta que es un sistema que va a generar una gran cantidad de datos debido a las diferentes funcionalidades del mismo, se pretende hacer uso de una base de datos que los almacene. Como se mencionó antes este sistema contará con distintas funcionalidades las cuales ayudarían a agilizar tareas como la generación de reportes de ventas por días, generación de corte diario, manejo de ventas por cliente, cobro por ventas, y el manejo del inventario.

La realización del estudio de métodos en la ferretería es importante debido a que partir de este, se puede realizar una mejora a los procesos estudiados, haciendo que sean más óptimos y que a su vez permitan aumentar los niveles de productividad de la compañía.

Capítulo 1

Propósito

El objetivo de este proyecto constituye una herramienta que permite un manejo oportuno y eficiente de la información, siendo posible el rendimiento de tiempo respecto a sus productos. Al igual que permita lograr la eficiencia en el control de los productos con los que se va a comercializar, como también llevar un control del inventario, registro de ventas en productos de una manera más rápida, organizada y sencilla beneficiando tanto al personal como a los clientes de la empresa. Emitirá reportes sobre ventas de manera eficaz, de forma que se logre un buen manejo de las transacciones que incurren en la empresa.

Capítulo 2

Ámbito del Proyecto

La ferretería a la cual está destinado el proyecto, desea un sistema de administración para ingreso y salida de mercancía sobre control del inventario y ventas, que ayude a mejorar la administración y control de la empresa, ya que al contar con un sistema manual se tienen dificultades para obtener de forma ágil y rápida la información relevante sobre las existencias. El proyecto consiste en diseñar y construir un sistema de interfaz amigable que permita llevar un control sobre los productos disponibles, además de mantener un registro de las salidas de los productos, de tal forma que esta información sirva al propietario del negocio para tomar decisiones como cuando reabastecerse de productos, también se pretende que el sistema gestione las ventas realizando su registro y generando un reporte de las mismas cuando este se requiera. Para el desarrollo de este proyecto se hará uso de distintas técnicas de programación aprendidas durante el curso como programación orientada a objetos, este paradigma nos servirá para la creación de la interfaz gráfica de usuario así como la funcionalidad deseada, combinando el lenguaje de programación Java con el lenguaje de consulta de mysql para poder manipular de mejor manera los datos.

Capítulo 3

Personal Involucrado

Nombre	Rol de usuario
Everardo de Jesus Pacheco Antonio	Administrador
Rolando Pedro Gabriel	Administrador
Lirio Ruiz Guerra	Empleado
Ramiro Ramírez Ramírez	Administrador
Leonel Alexis García Juárez	Administrador
Griselda Medina Avendaño	Administrador

Figura 3.1: Roles de usuario

Capítulo 4

Factibilidad

Es posible realizar este proyecto dado que para la factibilidad técnica, el equipo de desarrollo cuenta con conocimientos en programación en java así como el manejo de base de datos, además de que se cuenta con herramientas como el entorno de desarrollo Netbeans de cual se hará uso, además del sistema gestor de base de datos Postgres. En cuanto a la factibilidad operativa, se cuenta con usuarios finales dispuestos a emplear las funcionalidades generadas por el sistema que se desarrollará.

Capítulo 5

Viabilidad

Este proyecto es viable ya que contamos con conocimiento acerca de las distintas herramientas de desarrollo de software (Java, Postgres,MySQL). Además de esto cabe recalcar que contamos con las herramientas necesarias de trabajo, como Ide's para formar la base de nuestro sistema que está destinado a una ferretería. Se cuenta con el equipo necesario para lograr el objetivo. También nos encontramos organizados como equipo, y esto hace que el trabajo sea más sencillo de realizar, ya que cada uno aporta distintas ideas en cuanto a diseño, lógica y redacción de los distintos aspectos que requiera el sistema. El sistema será viable ya que se cuenta con los recursos necesarios que se requiera durante el proceso del mismo.

En cuanto al aspecto económico del sistema, no tenemos contemplado recurrir a gastos innecesarios para material externo, porque contamos con el material y herramientas adecuadas para la realización. La ventaja que se tiene es que cada integrante cuenta con un recurso material (laptos) para realizar las bases del sistema, por la parte de licencias de software o algún pago externo se ocupará software libre, gracias a que la herramienta Java es multiplataforma se evitarán costes para recurrir a esto.

Capítulo 6

Mapa de Navegación

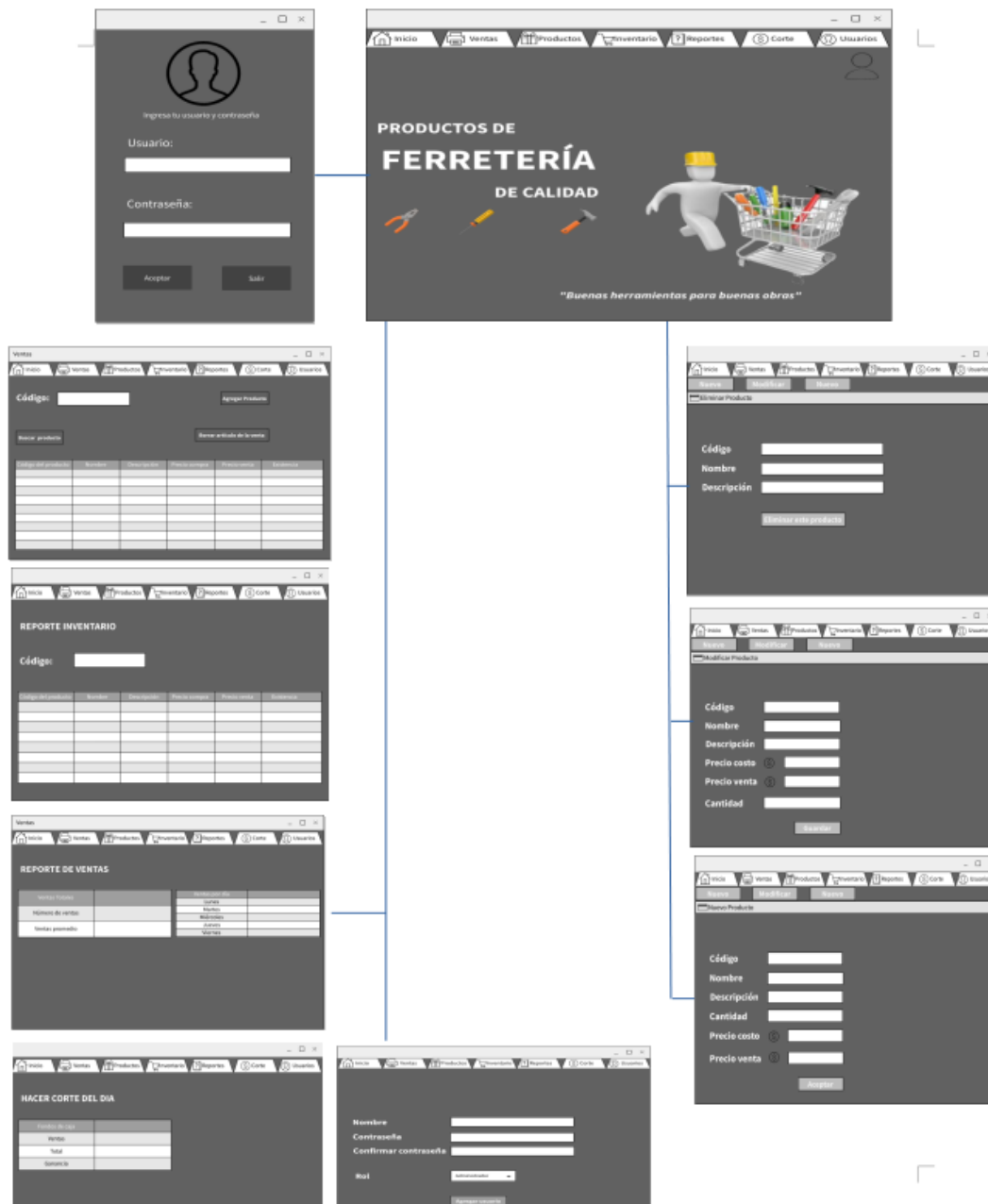


Figura 6.1: Mapa de navegación

Capítulo 7

Requisitos Funcionales

IDENTIFICADOR RF1		NOMBRE Buscar Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Producto		SALIDA Producto con todos sus datos (descripción, precio, existencia)	
DESCRIPCIÓN PRECONDICIÓN: Se desea localizar un producto por nombre o código. DESCRIPCIÓN: Se verifica el nombre del producto y el código. POSTCONDICIÓN: Se localiza el producto y se muestran sus datos.			
MANEJO DE SITUACIONES ANORMALES 1. Producto inexistente			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se encuentra el producto.			

Figura 7.1: Requisito Funcional 1

IDENTIFICADOR RF2		NOMBRE Agregar Producto a la venta	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Se agrega producto a la venta	
DESCRIPCIÓN PRECONDICIÓN: Se requiere localizar producto para agregar a la venta. DESCRIPCIÓN: Se busca producto deseado POSTCONDICIÓN: Se agrega producto localizado a la venta.			
MANEJO DE SITUACIONES ANORMALES 1.Producto inexistente, se envía mensaje de error al usuario.			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se localiza el producto.			

Figura 7.2: Requisito Funcional 2

IDENTIFICADOR RF3		NOMBRE Borrar articulo de la venta	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Selección del producto		SALIDA Producto eliminado de la venta	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con un producto de la venta seleccionado. DESCRIPCIÓN: Se consulta al usuario si quiere eliminar el artículo. POSTCONDICIÓN: Se elimina el producto de la venta.			
MANEJO DE SITUACIONES ANORMALES			
CRITERIOS DE ACEPTACIÓN 1. Se selecciona producto de la venta y se elimina de la misma.			

Figura 7.3: Requisito Funcional 3

IDENTIFICADOR RF4		NOMBRE Nuevo Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Cantidad Precio costo Precio venta		SALIDA Alta del producto en el sistema	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro del producto en sistema. DESCRIPCIÓN: Se ingresan los datos del producto. POSTCONDICIÓN: Producto es dado de alta en el sistema.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se reconoce el nuevo producto).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se agrega un nuevo producto.			

Figura 7.4: Requisito Funcional 4

IDENTIFICADOR RF5		NOMBRE Modificar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Precio Costo Precio venta Cantidad		SALIDA Datos de producto son modificados	
DESCRIPCIÓN PRECONDICIÓN: Se desea modificar algún dato del producto. DESCRIPCIÓN: Se ingresan los datos y el código del producto. POSTCONDICIÓN: Se modifica el producto.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se modifica el producto.			

Figura 7.5: Requisito Funcional 5

IDENTIFICADOR RF6		NOMBRE Eliminar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Producto eliminado del sistema	
DESCRIPCIÓN PRECONDICIÓN: Se desea dar de baja un producto, eliminar todos sus datos. DESCRIPCIÓN: Se ingresa código o nombre del producto para su localización. POSTCONDICIÓN: Se elimina producto del inventario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se elimina el producto.			

Figura 7.6: Requisito Funcional 6

IDENTIFICADOR RF7		NOMBRE Reporte de inventario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Inventario		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Productos almacenados en la base de datos		SALIDA Reporte de inventario.	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con productos dados de alta. DESCRIPCIÓN: Se ingresa a la sección inventario. POSTCONDICIÓN: Se genera un reporte de inventario donde se pueden ver todos los productos disponibles con sus datos correspondientes.			
MANEJO DE SITUACIONES ANORMALES 1. No se han dado de alta productos en sistema (Se mostrará en pantalla un mensaje de inexistencias).			
CRITERIOS DE ACEPTACIÓN 1. Existen productos dados de alta, se genera el reporte de inventario.			

Figura 7.7: Requisito Funcional 7

IDENTIFICADOR RF8		NOMBRE Reporte de ventas	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección reportes		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Ventas totales realizadas en una semana		SALIDA Reporte de: Ventas totales Número de ventas Venta promedio Ventas por día	
DESCRIPCIÓN PRECONDICIÓN: Se han realizado ventas DESCRIPCIÓN: Se consulta el importe total generado por cada venta. POSTCONDICIÓN: Se genera el reporte de ventas.			
MANEJO DE SITUACIONES ANORMALES 1. No se han realizado ninguna venta en la semana(Se mostrará en pantalla un mensaje).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas, se genera el reporte de ventas.			

Figura 7.8: Requisito Funcional 8

IDENTIFICADOR RF9		NOMBRE Hacer corte del día	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección corte		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Importe total de las ventas realizadas en el día Fondo de caja inicial		SALIDA Visualización de: Fondo de caja Ventas Total Ganancia	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con ventas realizadas en el día. DESCRIPCIÓN: Se realizan cálculos para mostrar total de ventas y ganancias. POSTCONDICIÓN: Se genera el corte del día.			
MANEJO DE SITUACIONES ANORMALES 1. No se ha realizado ni una venta en el día (Se mostrará en pantalla un mensaje al usuario).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas en el día, se genera corte del día.			

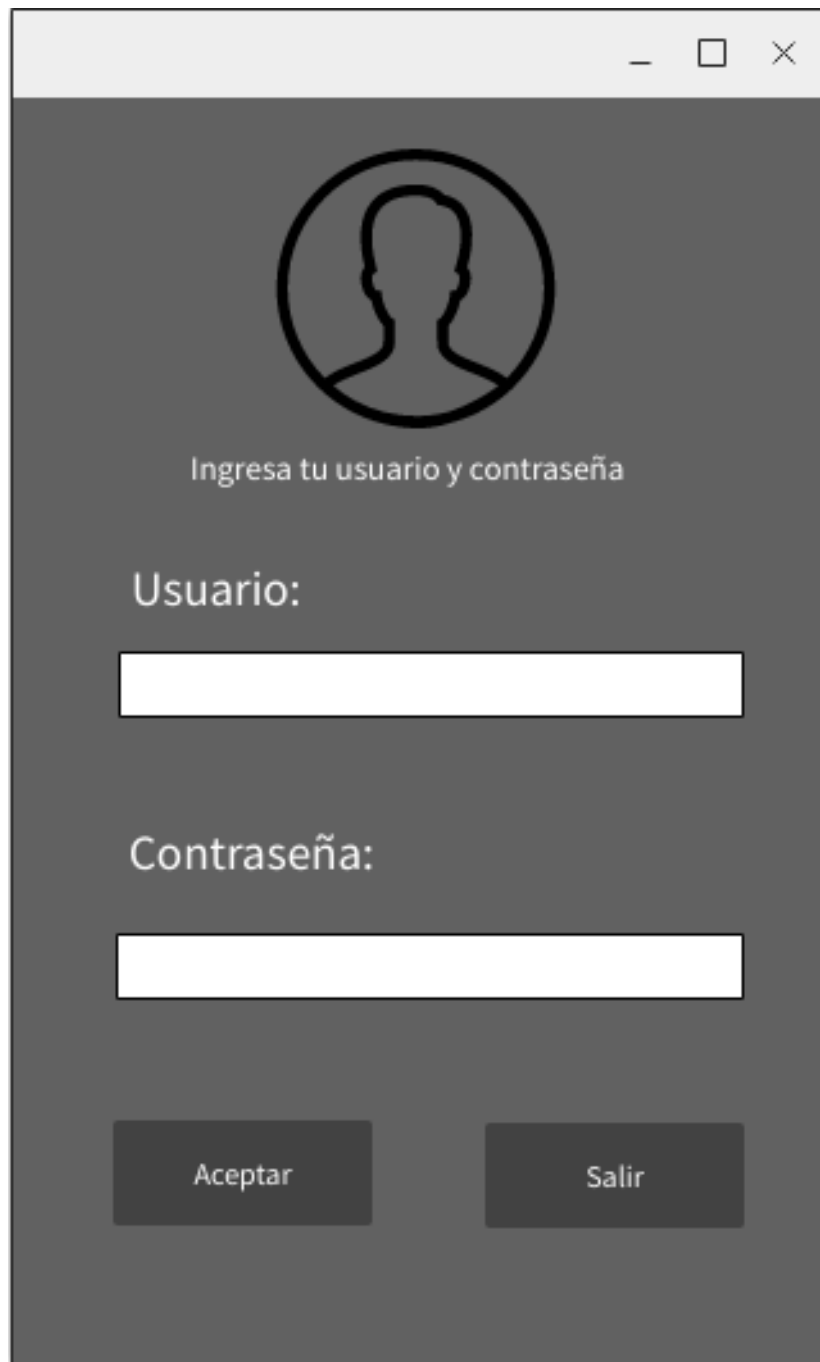
Figura 7.9: Requisito Funcional 9

IDENTIFICADOR RF10		NOMBRE Agregar Usuario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección Usuarios		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre Contraseña Rol		SALIDA Registro de nuevo usuario.	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro de usuario en el sistema. DESCRIPCIÓN: Se ingresan los datos del usuario a registrar. POSTCONDICIÓN: Se registra el nuevo usuario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se ingresa el usuario).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se registra un nuevo usuario.			

Figura 7.10: Requisito Funcional 10

Capítulo 8

Interfaces



A login window with a dark gray background and a light gray title bar. The title bar contains standard window controls (minimize, maximize, close). The main content area features a circular icon of a person's silhouette. Below the icon is the text "Ingresa tu usuario y contraseña". There are two input fields: "Usuario:" and "Contraseña:". At the bottom, there are two buttons: "Aceptar" and "Salir".

Ingresa tu usuario y contraseña

Usuario:

Contraseña:

Aceptar Salir

Figura 8.1: Login



Figura 8.2: Pantalla Principal

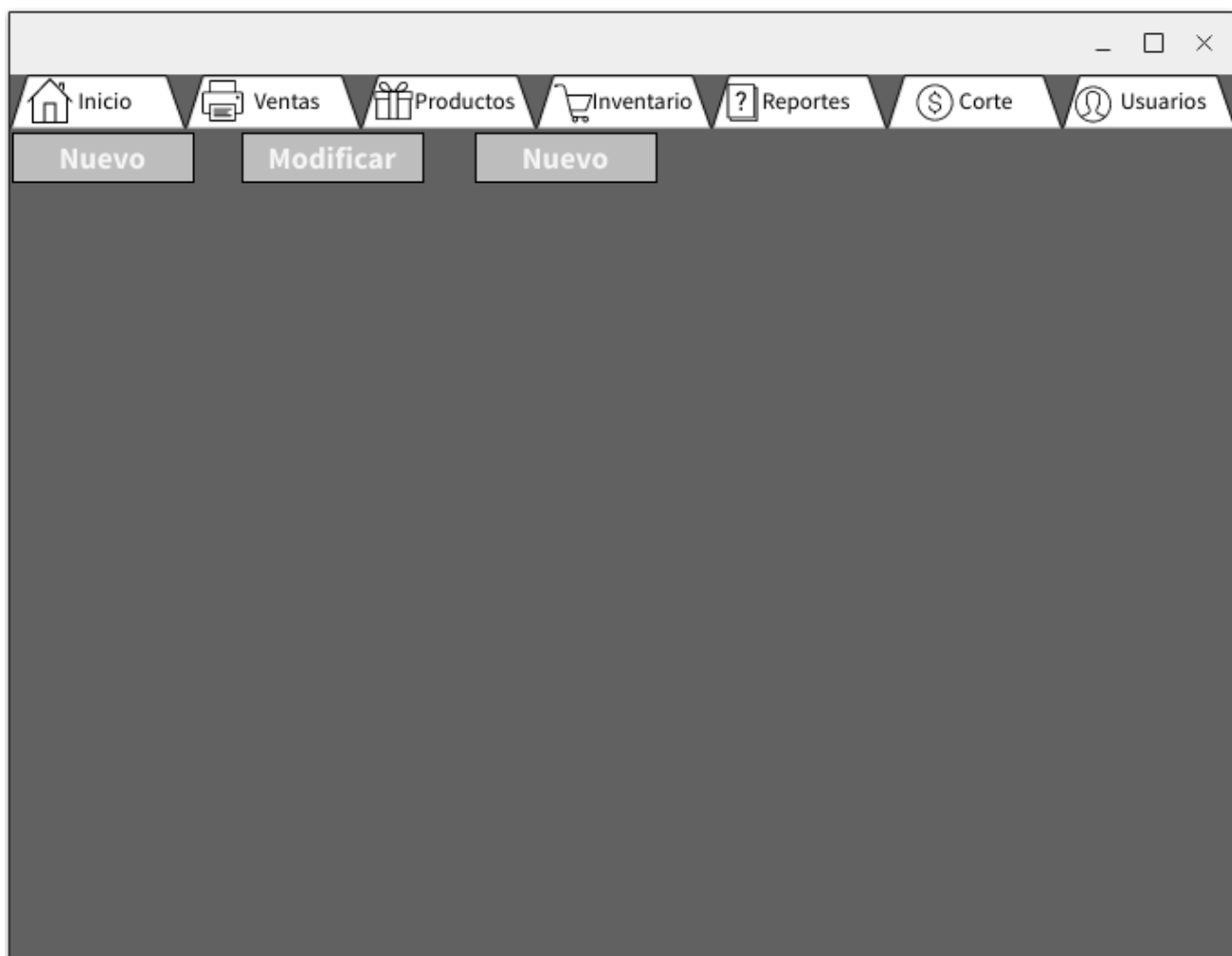


Figura 8.3: Productos

Inicio Ventas Productos Inventario Reportes Corte Usuarios

Nuevo Modificar Nuevo

Nuevo Producto

Código

Nombre

Descripción

Cantidad

Precio costo \$

Precio venta \$

Aceptar

Figura 8.4: Nuevo Producto

The screenshot shows a web application interface for managing products. At the top, there is a navigation bar with icons and labels for 'Inicio', 'Ventas', 'Productos', 'Inventario', 'Reportes', 'Corte', and 'Usuarios'. Below this bar are three buttons: 'Nuevo', 'Modificar', and 'Nuevo'. The main content area is titled 'Modificar Producto' and contains the following form fields:

- Código**: A text input field.
- Nombre**: A text input field.
- Descripción**: A text input field.
- Precio costo**: A text input field with a dollar sign icon (\$).
- Precio venta**: A text input field with a dollar sign icon (\$).
- Cantidad**: A text input field.

A 'Guardar' button is located at the bottom right of the form.

Figura 8.5: Modificar Producto

The screenshot shows a web application window with a standard browser title bar (minimize, maximize, close buttons). The application has a top navigation bar with icons and labels for 'Inicio' (home), 'Ventas' (sales), 'Productos' (products), 'Inventario' (inventory), 'Reportes' (reports), 'Corte' (cut), and 'Usuarios' (users). Below this is a secondary bar with three buttons: 'Nuevo' (New), 'Modificar' (Modify), and 'Nuevo' (New). A third bar contains a tab icon and the text 'Eliminar Producto'. The main content area has a dark gray background and contains three white input fields labeled 'Código', 'Nombre', and 'Descripción'. Below these fields is a button labeled 'Eliminar este producto'.

Inicio Ventas Productos Inventario Reportes Corte Usuarios

Nuevo Modificar Nuevo

Eliminar Producto

Código

Nombre

Descripción

Eliminar este producto

Figura 8.6: Eliminar Producto

The screenshot shows a web application window with a navigation bar at the top containing icons and labels for 'Inicio', 'Ventas', 'Productos', 'Inventario', 'Reportes', 'Corte', and 'Usuarios'. The main content area is titled 'REPORTE INVENTARIO' and features a 'Código:' label followed by a text input field. Below this is a table with six columns: 'Código del producto', 'Nombre', 'Descripción', 'Precio compra', 'Precio venta', and 'Existencia'. The table contains ten rows, with alternating light gray and white background colors for each row.

Código del producto	Nombre	Descripción	Precio compra	Precio venta	Existencia

Figura 8.7: Inventario

Ventas

Inicio

Ventas

Productos

Inventario

Reportes

Corte

Usuarios

Código:

Agregar Producto

Buscar producto

Borrar artículo de la venta

Código del producto	Nombre	Descripción	Precio compra	Precio venta	Existencia

Figura 8.8: Ventas



Figura 8.9: Reporte



The screenshot shows a web application interface with a navigation bar at the top. The navigation bar contains seven items: 'Inicio' (home icon), 'Ventas' (printer icon), 'Productos' (gift icon), 'Inventario' (shopping cart icon), 'Reportes' (question mark icon), 'Corte' (dollar sign icon), and 'Usuarios' (user icon). The 'Corte' item is highlighted. Below the navigation bar, the main content area has a dark gray background. It features the title 'HACER CORTE DEL DIA' in white text. Below the title is a table with four rows and two columns. The first column contains the labels 'Fondos de caja', 'Ventas', 'Total', and 'Ganancia'. The second column is empty, representing input fields for the values.

Fondos de caja	
Ventas	
Total	
Ganancia	

Figura 8.10: Corte

The image shows a web application window with a navigation bar at the top. The navigation bar contains seven items: 'Inicio' (home icon), 'Ventas' (printer icon), 'Productos' (gift icon), 'Inventario' (shopping cart icon), 'Reportes' (question mark icon), 'Corte' (dollar sign icon), and 'Usuarios' (user icon). The main content area is dark gray and contains a form for adding a user. The form has four labels: 'Nombre', 'Contraseña', 'Confirmar contraseña', and 'Rol'. Each label is followed by a white input field. The 'Rol' field is a dropdown menu with 'Administrador' selected. Below the 'Rol' field is a button labeled 'Agregar usuario'.

Inicio Ventas Productos Inventario Reportes Corte Usuarios

Nombre

Contraseña

Confirmar contraseña

Rol

Figura 8.11: Usuarios

Capítulo 9

Estándar de Codificación

[T1]fontenc[utf8]inputenc

9.1. Inicialización

Toda variable local deberá ser inicializada en el momento de declaración, deberá de estar situada al principio de cada bloque principal.

9.2. Variables locales

Deben estar situadas en el principio de cada bloque principal:

```
public class Ejemplo
    public static void main(String[] args)
        //Declaración de variables
        int contador=0;
        double r=0.0;
```

9.3. Variables globales

Estas deben estar ubicadas al inicio de la clase principal.

Notación

Se usará la notación CamelCase para la declaración de métodos o variables. //ejemplo de un método definido con CamelCase en CalcularSuma public int CalcularSuma(int b, int a) int resultado; resultado=b+a; return resultado;

9.4. Constantes

La declaración de contantes deberá ser declarado con letras mayúsculas.

```
public class Ejemplo
    public static void main(String[] args)
        //Declaración de constantes
        static final int DIASSEMANA = 7;
```

```
static final int DIASLABORABLES = 5;
```

9.5. Comentarios de inicio o encabezado

Se usará un encabezado al inicio de cada bloque o clase creada, también deberá ser asignada a las interfaces. El encabezado usado hará uso de comentarios en bloque como se muestra a continuación:

```
/** @Autor: */
/* Descripción del proyecto: */
/* Fecha de creación: */
/* Fecha de modificación: */
package ejemplo;
public class Ejemplo
public static void main(String[] args)
//Código
```

9.6. Comentarios en línea

Estos comentarios se deberán hacer en una línea, con cierto espacio si se quiere documentar la función de alguna operación o método.

```
public int CalcularSuma(int b, int a)
int resultado; //Se declara una variable de tipo entero
resultado=b+a; //Calcula la suma de b+a y la guarda en la variable resultado
return resultado; //Retorna la variable resultado
```

9.7. Sangrías y tabulaciones

Las tabulaciones serán igual a cuatro espacios, mientras que las sangrías deberán de mostrar claramente los procesos mas internos del código.

```
public class Ejemplo
    public static void main(String[] args)
        int num=0;
        if(num==0)
            System.out.println("Número igual a cero");

        else if(num>0)
            System.out.println("Número mayor que cero");

        else
            System.out.println("Número menor que cero");
```

9.8. Interfaces

Para los nombres que serán usados en los componentes de las interfaces se agregará un identificador, esto con el fin de reconocer mas fácil a los componentes. Los componentes que se usaran serán los

siguientes:

```
jTextField txtEjemplo  
jLabel lblEjemplo  
jButton btnEjemplo  
jList lsEjemplo  
jRadioButton rbtnEjemplo  
jCheckBox cbEjemplo  
jComboBox cbxEjemplo  
jPanel jpEjemplo  
jTabbedPane jtEjemplo  
jPasswordField jpsEjemplo  
jTable tbEjemplo
```

Capítulo 10

Diagramas de casos de uso

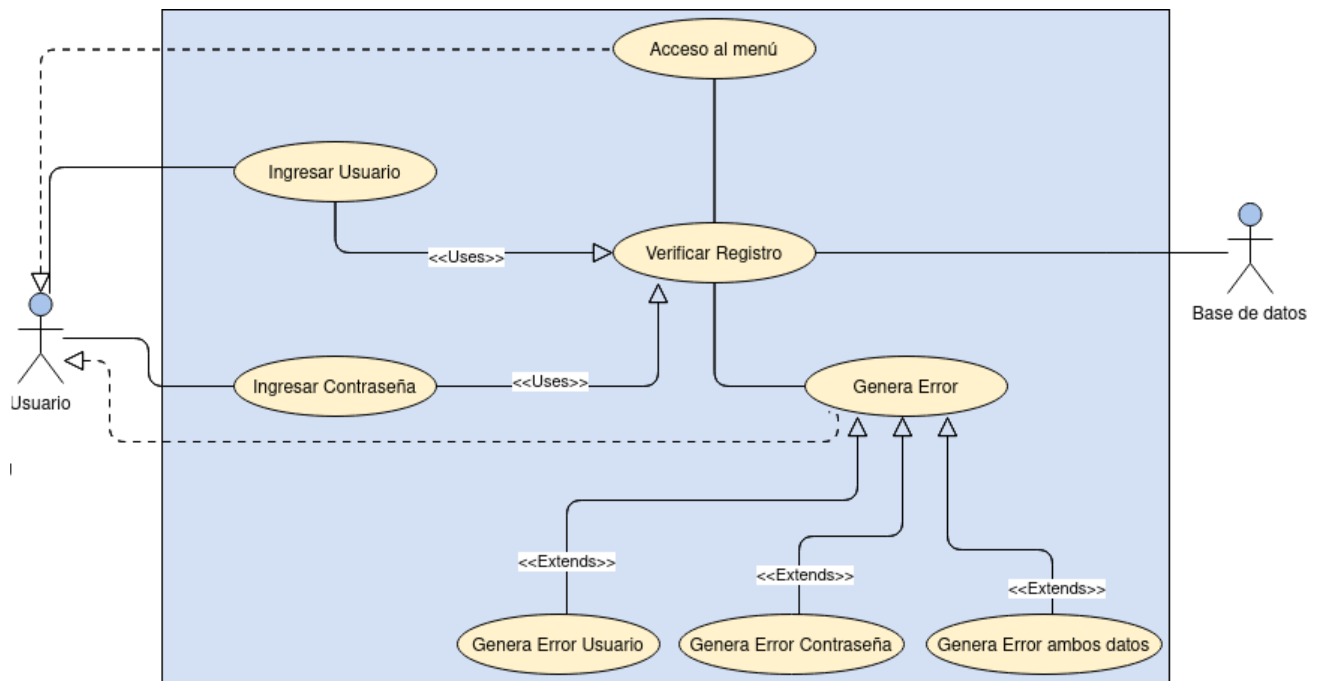


Diagrama de CU Login

ID:	CU-1		
Nombre:	Login		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, Usuario, Base de datos		
Descripción:	Conceder acceso del usuario al sistema.		
Disparador:	N/A		
Precondiciones:	N/A		
Postcondiciones:	Se accede a la pantalla inicio.		
Flujo normal:	<p>1. Ingresar nombre.</p> <p>2. Ingresar contraseña.</p> <p>3. Se presiona el botón aceptar.</p> <p>3.1 En caso de que nombre de usuario no exista se envía mensaje de error “Usuario erróneo”.</p> <p>3.2 En caso de que contraseña sea incorrecta se envía mensaje de error: “Contraseña incorrecta”.</p> <p>3.3 Si no se ha ingresado uno de los dos datos (nombre o contraseña) se envía mensaje de error: “Uno de los campos esta vacío”.</p> <p>2. El sistema valida los datos ingresados en 1 y 2.</p> <p>2.1 En caso de que los datos sean correctos se concede el acceso, y se remite a la pantalla “Inicio”.</p> <p>2.1 En caso de datos inválidos (nombre o contraseña) se envía mensaje de error al usuario.</p>		
Flujo alterno:	Una vez que se concede el acceso, usuario accede a la pantalla “inicio” con el menú de opciones que le corresponda según su rol(administrador o usuario).		
Includes:	N/A		
Frecuencia de uso:	Alta		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.1: CU Login

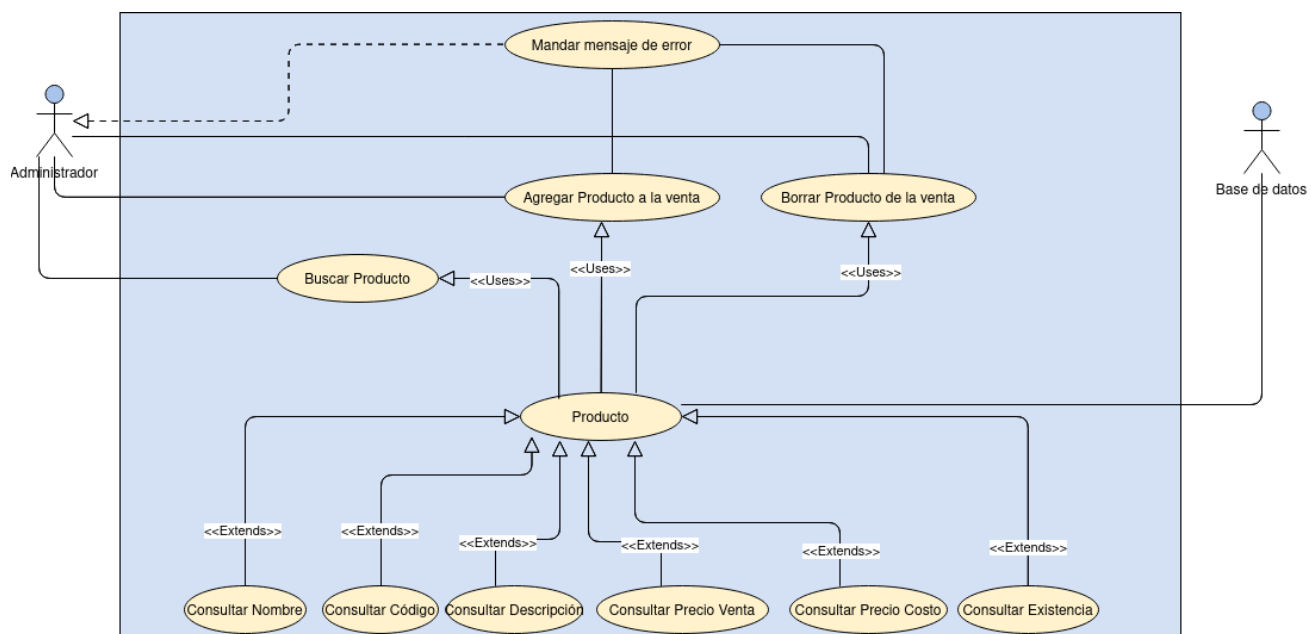


Figura 10.2: Diagrama de CU Ventas

ID	CU-2		
Nombre	Ventas de producto		
Creado por	RRR	Actualizado por:	RRR
Fecha de creación	30/10/2021	Fecha de última revisión:	30/10/2021
Actores	Administrador, usuario, base de datos		
Descripción	Registrar ventas y cobrar (Pantalla).		
Disparador	Se selecciona la opción “Ventas” del menú principal		
Precondiciones	Usuario debe haberse logeado en el sistema.		
Postcondiciones	Se reinician los valores y se muestra la pantalla ().		
Flujo normal	<ol style="list-style-type: none"> 1. Ingresar el código de barras del producto. <ol style="list-style-type: none"> 1.1 Si el código no existe, se envía un mensaje de “Error” y vuelve al punto 1, para ingresar un nuevo código. 2. El sistema muestra las características del producto en una tabla. <ol style="list-style-type: none"> 2.1 Si el usuario desea registrar otra venta ir al punto 1. <ol style="list-style-type: none"> 2.1.1 Se continua con el flujo. 3. Presionar el botón cobrar. 		
Flujo alterno	Al terminar de agregar los productos ir a la pantalla ().		
Includes	N/A		
Frecuencia de uso	Muy frecuente		
Requerimientos especiales	N/A		
Supuestos	N/A		
Isues y Notas	N/A		

Figura 10.3: CU Ventas

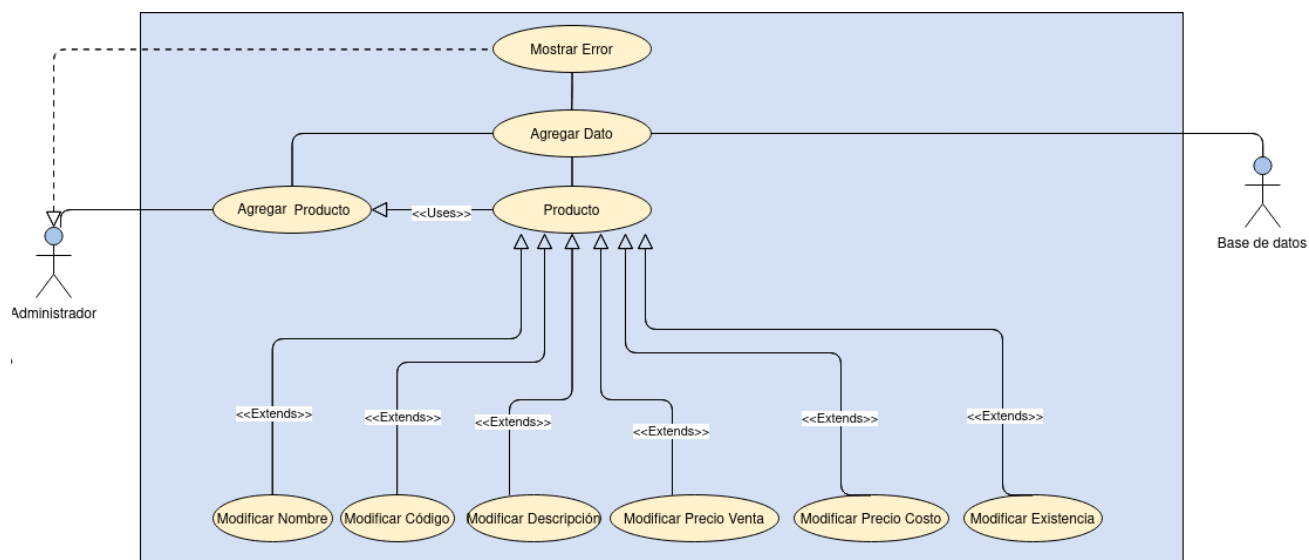


Figura 10.4: Diagrama de CU Nuevo Producto

ID:	CU-3		
Nombre:	Nuevo Producto		
Creado por:	RRR	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	03/12/2021
Actores:	Administrador, usuario, base de datos		
Descripción:	Registra nuevos productos con sus datos correspondientes en el sistema.		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Nuevo” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca la pantalla para que el usuario agregue nuevo producto si lo desea.		
Flujo normal	<p>1. Se ingresan datos del producto: Nombre, descripción, precio de compra, precio de venta, cantidad.</p> <p>1.1 El sistema valida los datos; el nombre y descripción sean ingresados con caracteres alfabéticos, en caso de ser diferentes no permite el ingreso de datos.</p> <p>1.2 El sistema valida que el precio de compra y precio de venta sean valores float, en caso de ser diferentes no se permite el ingreso de los datos.</p> <p>1.3 El sistema valida que la cantidad sea un valor entero en caso de ser diferente no se permite en ingreso de datos en pantalla.</p> <p>2. Pulsa el botón aceptar y se guarda el producto en la base de datos.</p> <p>2.1 En caso de que un campo este vacío se envía mensaje de error: “Uno de los campos esta vacío”.</p>		
Flujo alterno:	Al terminar de agregar los productos se refresca pantalla.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.5: CU Nuevo Producto

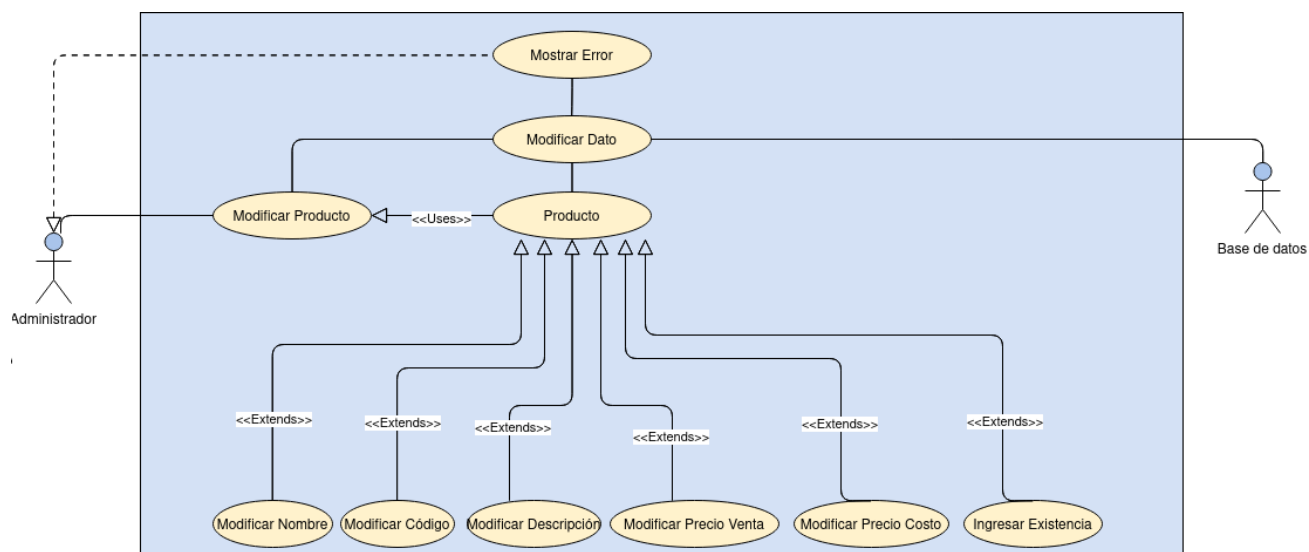


Figura 10.6: Diagrama de CU Modificar Producto

ID:	CU-4		
Nombre:	Modificar producto		
Creado por:	MAG	Actualizado por:	MAG
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, usuario, base de datos		
Descripción:	Realiza cambios a los datos del producto.		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Modificar” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se actualiza la información del producto en la base de datos.		
Flujo normal:	<ol style="list-style-type: none"> 1. Se ingresa código del producto. 2. Se muestran en pantalla los datos del producto que coincidan con el código ingresado. 3. Usuario edita campo del producto que desea actualizar. <ol style="list-style-type: none"> 3.1 Si los datos del nombre y descripción no corresponden a caracteres alfabéticos, no se permite el ingreso de los datos. 3.2 Si los datos precio compra y precio venta no corresponden a valores float no se permite el ingreso de los datos. 3.3 Si el dato cantidad no corresponde a un valor entero no se permite el ingreso de dicho dato. 4. Una vez que el usuario ha ingresado todos los datos que desea cambiar selecciona la opción “guardar”. 5. Se actualizan datos en la base de datos. 		
Flujo alterno:	Al terminar de agregar los productos se refresca la pantalla.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.7: CU Modificar Producto

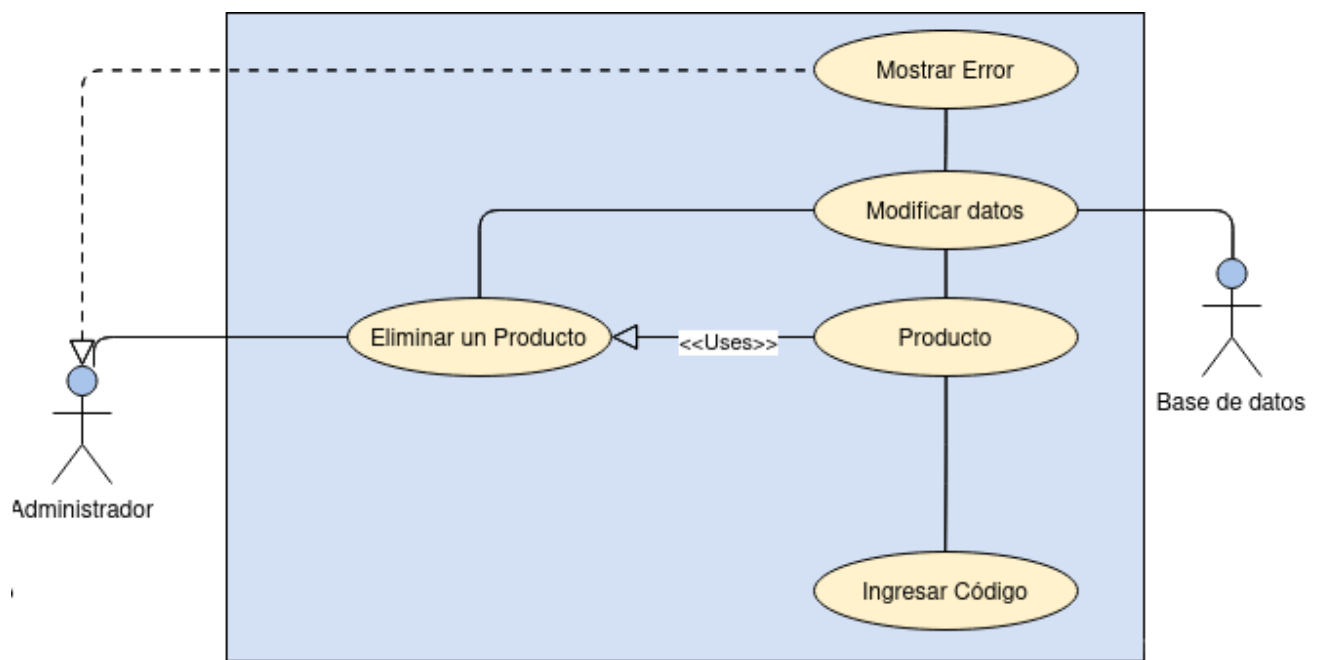


Figura 10.8: Diagrama de CU Eliminar Producto

ID:	CU-5		
Nombre:	Eliminar producto		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	01/12/2021
Actores:	Administrador, Usuario, Base de datos		
Descripción:	Elimina producto del sistema,		
Disparador:	Se selecciona la opción “Productos” del menú y enseguida presionar el botón “Eliminar” .		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca la pantalla.		
Flujo normal:	1. Se ingresa código de producto que desea eliminar. 2. El sistema muestra en pantalla los datos nombre y descripción del producto que coincida con el código ingresado. 3. Usuario presiona botón “eliminar” y se hace borrado lógico del producto en la base de datos.		
Flujo alterno:	Se refresca la pantalla “Eliminar producto” , para que usuario pueda eliminar otro producto si lo desea.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.9: CU Eliminar Producto

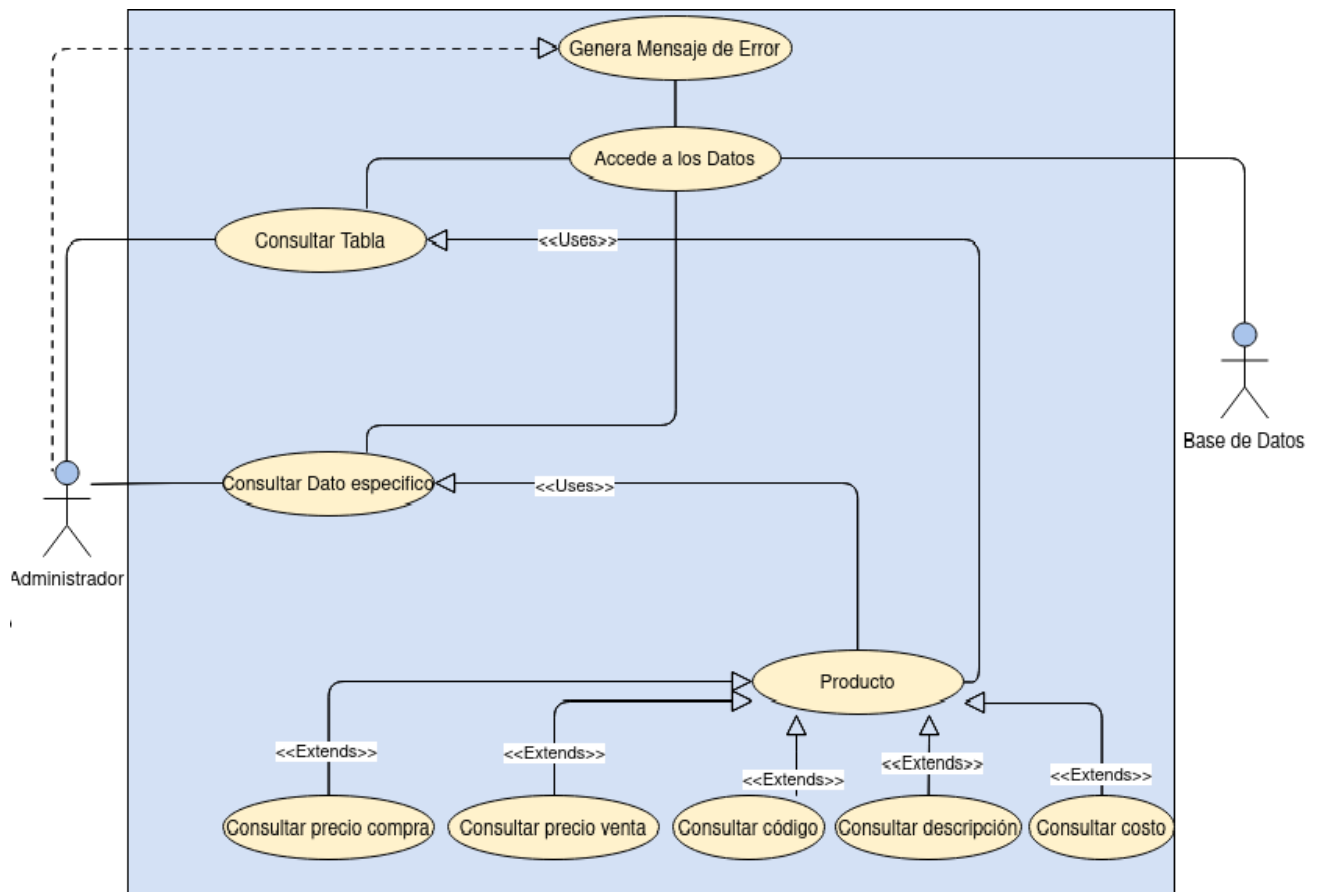


Figura 10.10: Diagrama de CU Inventario

ID:	CU-6		
Nombre:	Inventario		
Creado por:	MAG	Actualizado por:	MAG
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Muestra artículos registrados en el sistema.		
Disparador:	Se selecciona la opción “Inventario” del menú principal		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Al seleccionar opción “Inventario” del menú, el sistema muestra en una tabla los productos existentes en la base de datos. 2. Se ingresa código para ver datos de un producto en específico.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.11: CU Inventario

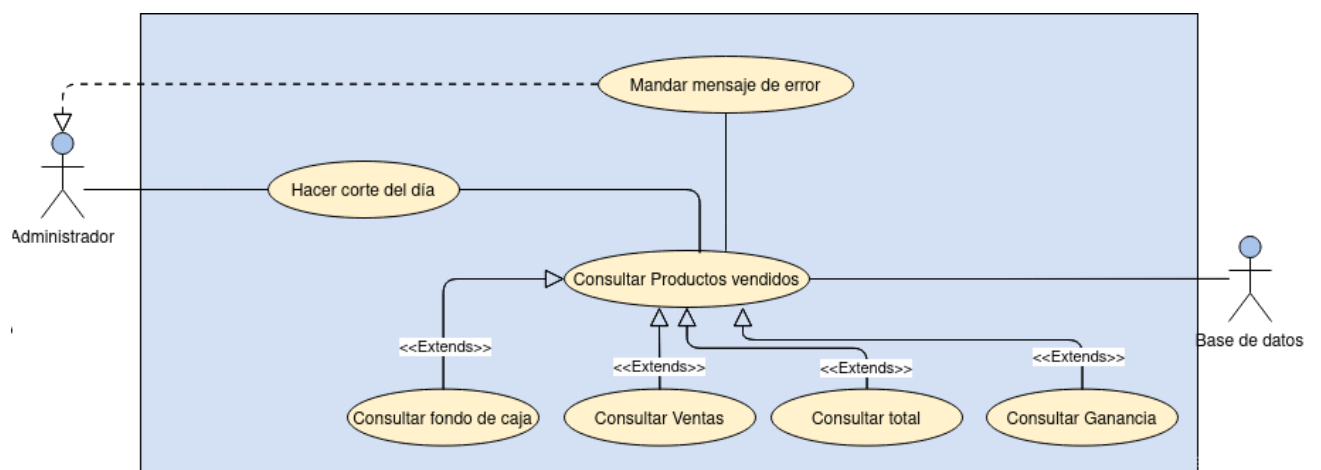


Figura 10.12: Diagrama de CU Corte

ID:	CU-7		
Nombre:	Generar corte de caja		
Creado por:	GJLA	Actualizado por:	GJLA
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, usuario, base de datos.		
Descripción:	Genera corte de caja del día, mostrando datos como Fondo de caja, Ventas (importe total de las ventas del día), total (dinero total en caja) y Ganancia.		
Disparador:	Se selecciona la opción “Corte” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Pulsa el botón “Hacer corte del día”. 2. El sistema muestra en una tabla la siguiente información: Fondo de caja, Ventas, Total y Ganancia.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.13: CU Corte

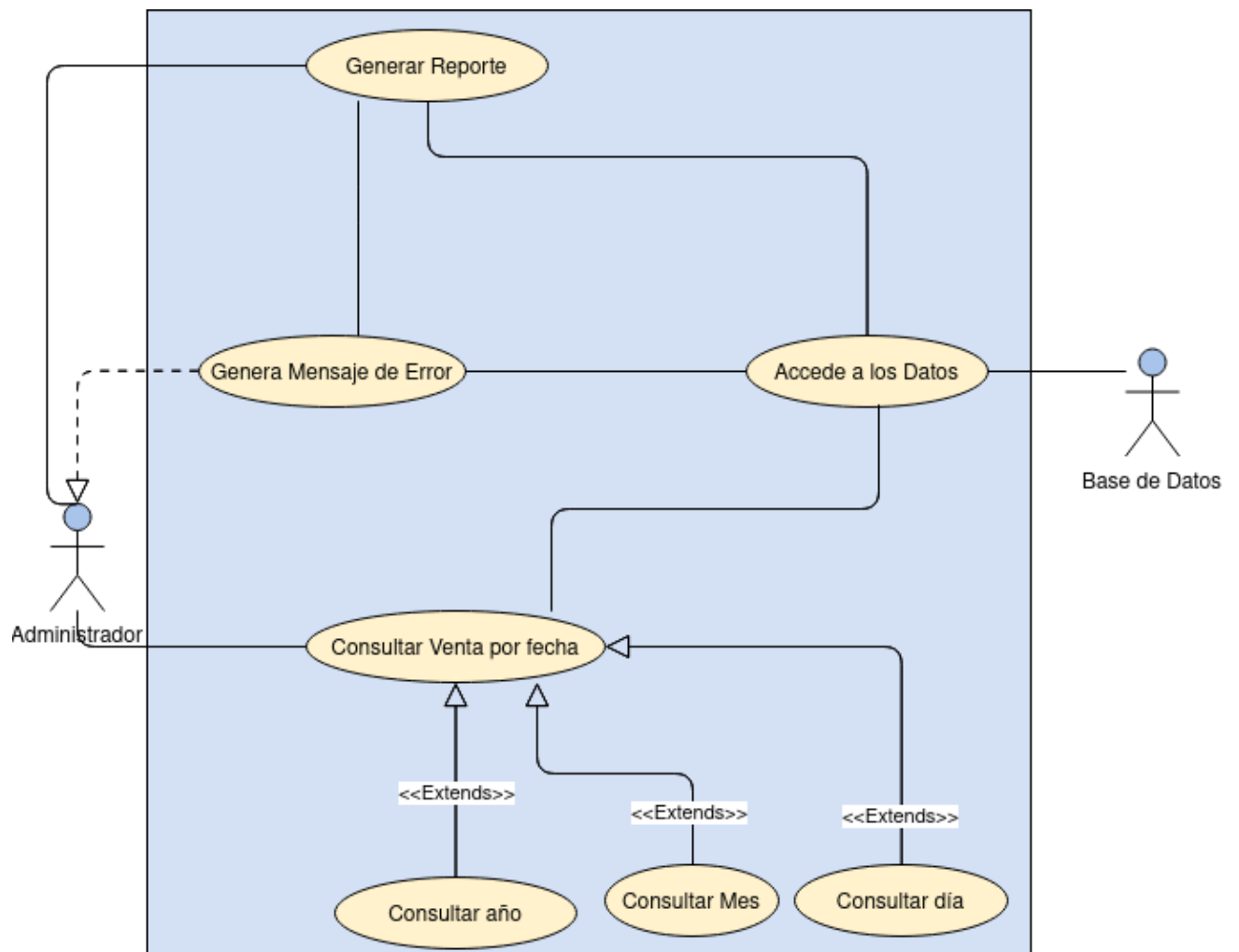


Figura 10.14: Diagrama de CU Reporte

ID:	CU-8		
Nombre:	Generar reporte		
Creado por:	RRR	Actualizado por:	RRR
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, base de datos.		
Descripción:	Genera reporte de ventas por día.		
Disparador:	Se selecciona la opción “Reporte” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	N/A		
Flujo normal:	1. Se pulsa opción “Reporte” del menú de opciones y se muestra en una tabla el resumen de ventas por día de la semana, así como las ventas totales del día, el número de ventas realizadas así como el importe por venta promedio.		
Flujo alterno:	Usuario puede dirigirse a cualquier opción del menú.		
Includes:	N/A		
Frecuencia de uso:	Baja		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.15: CU Reporte

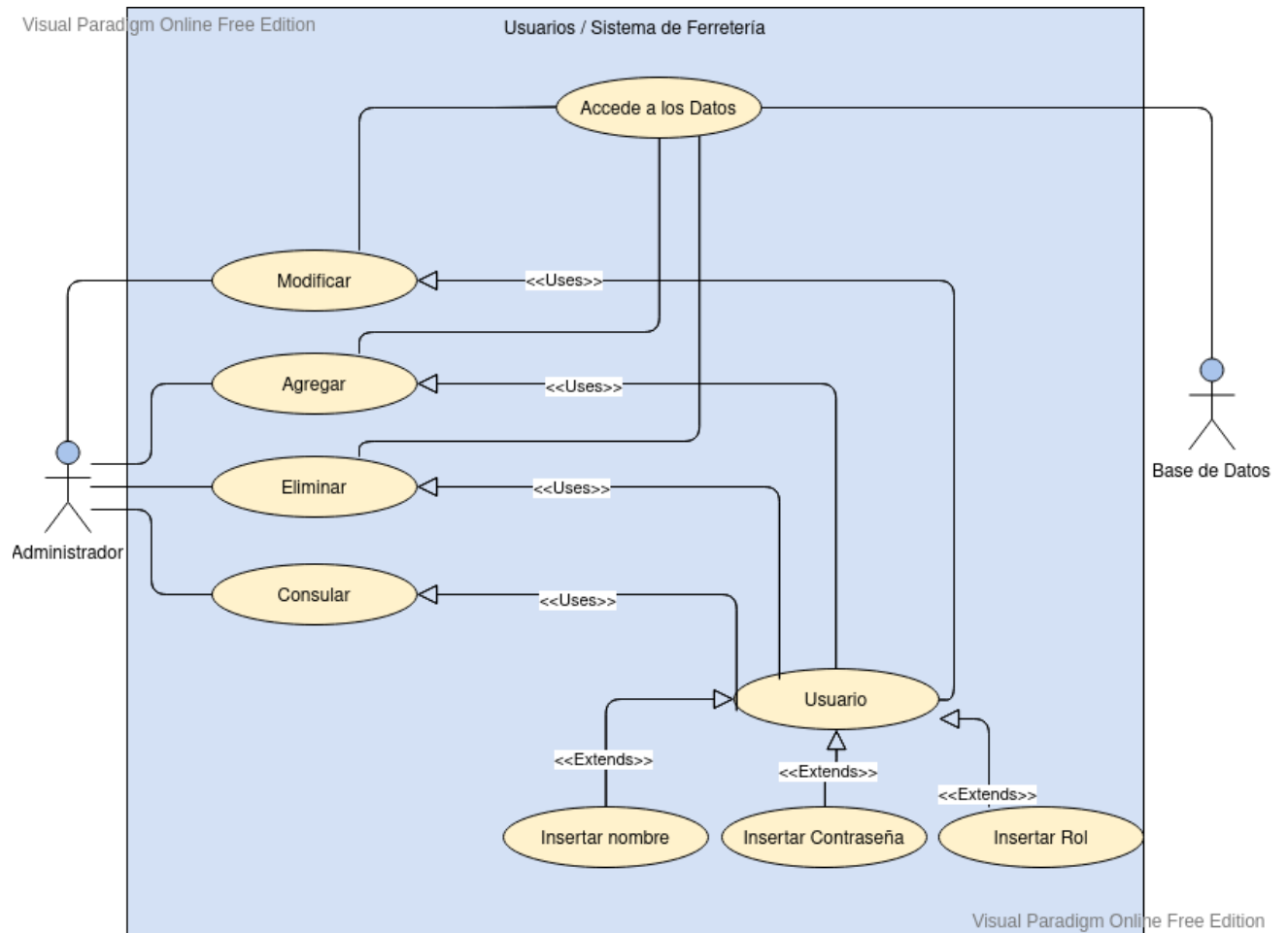


Figura 10.16: Diagrama de CU Usuarios

ID:	CU-9		
Nombre:	Agregar usuario		
Creado por:	RRR	Actualizado por:	RRR
Fecha de creación:	30/10/2021	Fecha de última revisión:	30/10/2021
Actores:	Administrador, base de datos.		
Descripción:	Registra nuevos usuarios en el sistema.		
Disparador:	Se selecciona la opción “Usuarios” del menú de opciones.		
Precondiciones:	Usuario debe haberse logeado en el sistema.		
Postcondiciones:	Se refresca pantalla, para que administrador pueda registrar un nuevo usuario si lo desea.		
Flujo normal:	1. Se ingresan datos del nuevo usuario a registrar: Nombre, Contraseña y rol (administrador o usuario). 2. Pulsa botón “Agregar usuario”. 3. Se registra nuevo usuario en la base de datos.		
Flujo alterno:	Se refresca la pantalla “Usuarios” , para que administrador pueda dar de alta a otro usuario si lo desea.		
Includes:	N/A		
Frecuencia de uso:	Muy frecuente		
Requerimientos especiales:	N/A		
Supuestos:	N/A		
Isues y Notas:	N/A		

Figura 10.17: CU Usuario

Capítulo 11

Diccionario de datos

En la tabla 1 se muestran los tipos de datos correspondientes a los productos ya que al registrar un producto es necesario tener un código que lo identifique, un nombre con una breve descripción, sus correspondientes precios(venta y compra) para los futuros cálculos. Así como se debe llevar un control de la Existencia del producto y su marca.

Columna		Tipo de dato	Descripción
PK	idCodigo	idCod	Almacena el código único del producto
	nombre	Varchar(25)	Se guarda el nombre del producto
	descripcion	Varchar(50)	Se guarda una breve descripcion acerca de las cualidades del producto
	precioVenta	flotante	Almacena el precio por el cual se vende el producto correspondiente
	precioCompra	flotante	Almacena el precio por el cual se compra el producto correspondiente
	Existencia	flotante	Se guarda un total del total de productos correspondientes
	Marca	Varchar(30)	Se guarda la marca del producto

Figura 11.1: Productos

En la tabla 2 Se muestran los Datos que se ocuparan para las ventas realizadas, se debe contar con un identificador de venta para la generación de reportes a futuro, en la descripción de venta se registraran los productos que formaron parte de esta venta, el total de venta y la fecha en la que se realizó.

La Tabla 3 nos muestra las categorías en esta tabla se guardaran contando con un id de la categoría para evitar que se duplique, así como las distintas categorías (Equipamiento, cerrajería, fontanería, materiales eléctricos, herramientas, etc), cuenta ademas con una descripción breve de esta misma.

Columna	Tipo de dato	Descripción
PK idVenta	idV	Identificador de la venta realizada
descripcion	varchar(50)	Se registraran los productos adquiridos por el cliente así como el total de ellos
totalVenta	flotante	Guarda el total de la venta realizada
fecha	varchar(10)	Almacena la fecha en la cual se realizó la venta

Figura 11.2: Ventas

Columna	Tipo de dato	Descripción
PK idCategoria	idCat	Identificador de la categoría del producto (Industrial, Hogar, etc)
nombre	Integer	Almacena el nombre de la categoría
descripcion	Integer	Se guarda una pequeña descripción del tipo de categoría que se hace referencia

Figura 11.3: Categorías

Columna	Tipo de dato	Descripción
PK idEmpleado	idE	Identificador del empleado
FK idVenta	idV	Identificador de la venta realizada
nombre	varchar(25)	Nombre del empleado
apellidos	varchar(25)	Apellidos del empleado
cargo	varchar(15)	Cargo que ocupa el empleado

Figura 11.4: Empleado

Columna	Tipo de dato	Descripción
FK idCodigo	idCod	Identificador del producto
FK idCategoria	idCat	Identificador de la categoría a la que pertenece el producto

Figura 11.5: Pertenece

Columna		Tipo de dato	Descripción
FK	idCodigo	idCod	Identificador del producto
FK	idVenta	idV	Identificador de la venta realizada

Figura 11.6: Registro

Capítulo 12

Diseño Lógico

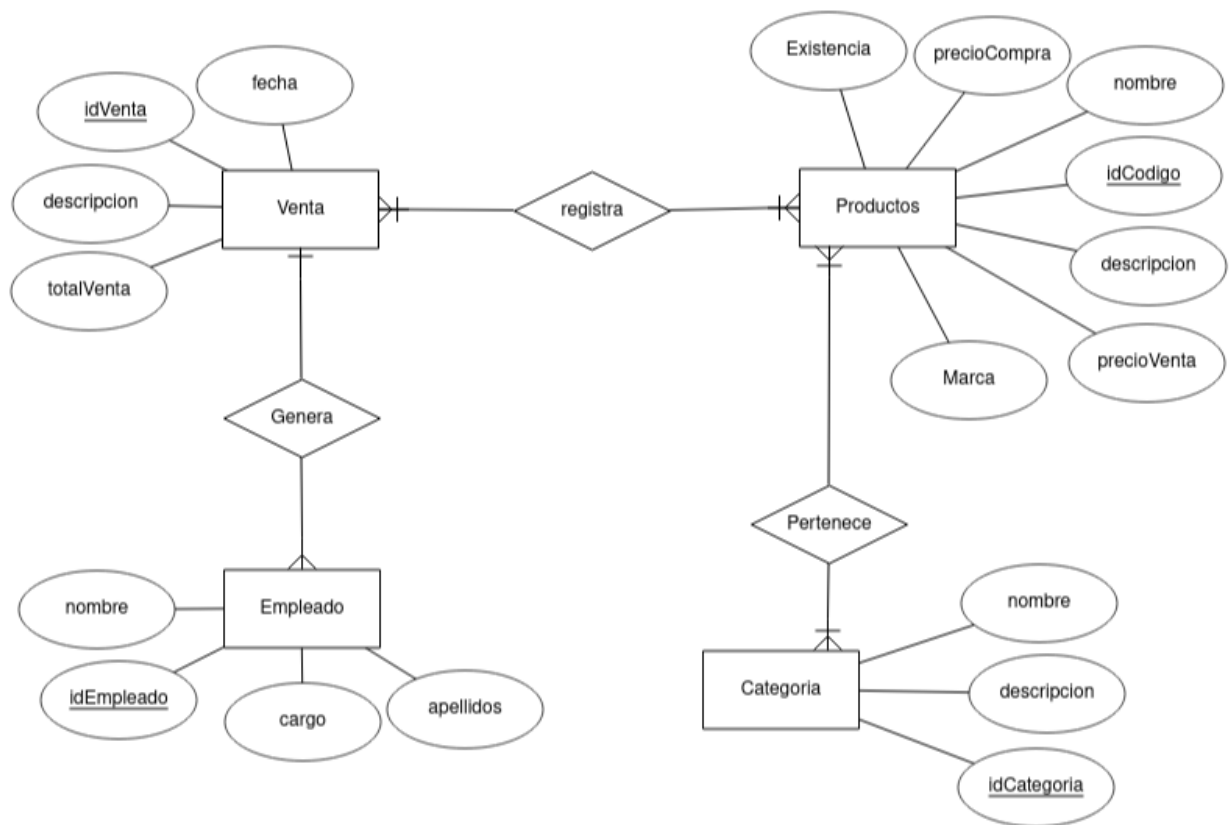


Figura 12.1: Diagrama ERD

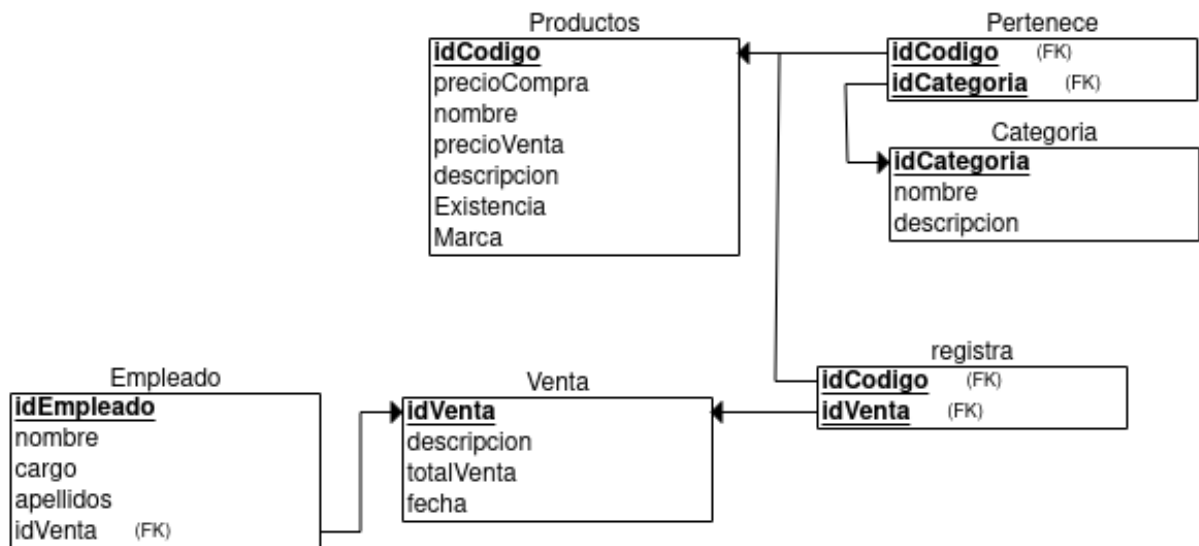


Figura 12.2: Esquema Relacional

Capítulo 13

Diseño Físico

```
create domain flotante as text constraint flotantecheckcheck(value 'l[0 - 9]1, 9.[0 - 9]0, 2'::text); create
domain idV as text constraint idVcheckcheck(value 'V[0 - 9]1, 9'::text); create domain idCod as text
constraint idCodcheckcheck(value 'P[0 - 9]1, 9'::text); create domain idCat as text constraint
idCatcheckcheck(value 'C[0 - 9]1, 9'::text); create domain idE as text constraint
idEcheckcheck(value 'E[0 - 9]1, 9'::text);
create table Productos(idCodigo idCod primary key, nombre varchar(25), descripcion varchar(50),
precioVenta flotante, precioCompra flotante, Existencia integer, Marca varchar(30));
create table Venta(idVenta idV primary key, descripcion varchar(50), totalVenta flotante, fecha
varchar(10));
create table Categoria(idCategoria idCat primary key, nombre varchar(25), descripcion varchar(50));
create table Empleado(idVenta idV, idEmpleado idE primary key, nombre varchar(25), apellidos
varchar(25), cargo varchar(15), FOREIGN KEY(idVenta)REFERENCES Venta(idVenta) ON
DELETE CASCADE ON UPDATE CASCADE);
create table Pertenece(idCodigo idCod, idCategoria idCat, FOREIGN
KEY(idCodigo)REFERENCES Productos(idCodigo) ON DELETE CASCADE ON UPDATE
CASCADE, FOREIGN KEY(idCategoria)REFERENCES Categoria(idCategoria) ON DELETE
CASCADE ON UPDATE CASCADE);
create table Registra(idCodigo idCod, idVenta idV, FOREIGN KEY(idCodigo)REFERENCES
Productos(idCodigo) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN
KEY(idVenta)REFERENCES Venta(idVenta) ON DELETE CASCADE ON UPDATE
CASCADE);create database ferregra; ferregra
create domain flotante as text constraint
flotantecheckcheck(value 'l[0 - 9]1, 9.[0 - 9]0, 2'::text); create domain idV as text constraint
idVcheckcheck(value 'V[0 - 9]1, 9'::text); create domain idCod as text constraint
idCodcheckcheck(value 'P[0 - 9]1, 9'::text); create domain idCat as text constraint
idCatcheckcheck(value 'C[0 - 9]1, 9'::text); create domain idE as text constraint
idEcheckcheck(value 'E[0 - 9]1, 9'::text);
create table Productos(idCodigo idCod primary key, nombre varchar(25), descripcion
varchar(50), precioVenta flotante, precioCompra flotante, Existencia integer, Marca
varchar(30));
create table Venta(idVenta idV primary key, descripcion varchar(50), totalVenta
flotante, fecha varchar(10));
create table Categoria(idCategoria idCat primary key, nombre varchar(25), descripcion
varchar(50));
```

```
create table Empleado(idVenta idV, idEmpleado idE primary key, nombre varchar(25),  
  apellidos varchar(25), cargo varchar(15), FOREIGN KEY(idVenta)REFERENCES  
    Venta(idVenta) ON DELETE CASCADE ON UPDATE CASCADE);  
  create table Pertenece(idCodigo idCod, idCategoria idCat, FOREIGN  
KEY(idCodigo)REFERENCES Productos(idCodigo) ON DELETE CASCADE ON  
  UPDATE CASCADE, FOREIGN KEY(idCategoria)REFERENCES  
    Categoria(idCategoria) ON DELETE CASCADE ON UPDATE CASCADE);  
  create table Registra(idCodigo idCod, idVenta idV, FOREIGN  
KEY(idCodigo)REFERENCES Productos(idCodigo) ON DELETE CASCADE ON  
UPDATE CASCADE, FOREIGN KEY(idVenta)REFERENCES Venta(idVenta) ON  
  DELETE CASCADE ON UPDATE CASCADE);
```